homework-001-report

November 24, 2019

1 Athens University of Economics and Business

2 M.Sc. in Data Science (part time)

Course: Social Network Analysis (INF322)

Instructor: Katia Papakonstantinopoulou, Dept. of Informatics

Semester: Fall 2019

Homework 1: SNAP High performance system for analysis and manipulation of large networks

Submission of: Spiros Politis (p3351814)

2.1 Euler Paths and Circuits

The first part of this homework requires that you familiarize yourselves with graph creation and graph traversal with SNAP.

You have to develop two Python functions that examine whether a given graph has:

- an Euler Path, and
- an Euler Circuit.

An Euler path is a path that uses every edge of a graph exactly once. An Euler path starts and ends at different vertices. If a graph has an Euler path, then it must have exactly two vertices with odd degree, and it is these odd vertices that will form the beginning and end of the path.

An Euler circuit is a circuit that uses every edge of a graph exactly once. An Euler circuit starts and ends at the same vertex. If a graph has an Euler circuit, then all of its vertices must be of even degree.

Note that you also must verify that the graph is connected for both cases, which can be easily done using SNAP: https://snap.stanford.edu/snappy/doc/reference/cncom.html.

In addition to implementing the above functions you should complete the test- case that is provided with this homework (hw1-1.py). In particular, you should complete all four tests by filling in code that creates graphs that satisfy the tests' assertions:

- A graph that has an Euler path (but not an Euler circuit),

- A graph that does not have an Euler path,
- A graph that has an Euler circuit, and
- A graph that does not have an Euler circuit.

Note that submitting your homework with tests that succeed does not guarantee full points in this exercise. You should make sure that both functions are correctly implemented, and the graphs in your tests are created appropriately.

Remember that if you wish to create a graph of significant size, you can employ one of the available graph generators:

https://snap.stanford.edu/snappy/doc/reference/generators.html.

2.1.1 Answer

Execution remarks In order to run the test cases included in file *hw1-1.py*, one should execute it in the following fashion:

```
python hw1-1.py 1000
```

where the first parameter is the number of nodes in the graph.

Code structure (package SnaHomework1.Part1) The code is structured in the following classes:

- **GraphGenerator**: contains all functions so as to generate the graphs required for each part of the assignment.
- **GraphEvaluator**: contains all functions so as to evaluate the questions of the assignment (e.g. a graph has an Euler path).
- **Util**: contains helper functions.

Note that code is fully documented with inline comments.

Output

```
....
----------------------------------------------------------------------
Ran 4 tests in 3.110s

OK
```

2.2 Apply node centrality measures and community detection algorithms on generated graphs

For the second part of this homework, you will have to write a Python script (hw1-2.py) that generates a graph of given size, reports some information on the graph, and compares the execution times of two community detection algorithms.

You will generate graphs using the Watts-Strogatz model. Start with a graph of 50 nodes and set the out-degree of each node to a value of your choice in [5, 20]. First, your script will print out the id of the node with the highest degree as well as its degree. Then, you will print out the ids of the nodes with the highest Hub and Authority scores as well as their scores. Finally, you will measure the time needed for the execution of the Girvan-Newman community detection algorithm based on betweenness centrality and the Clauset-Newman-Moore community detection method.

Your task is to execute this script multiple times by increasing the number of nodes parameter, to report the execution time for graphs of different sizes. You should repeat execution to the point that both algorithms require more than 10 minutes to execute or you receive a memory error.

Finally, for the largest among the graphs you will generate, find the top-30 nodes of highest PageRank and compute the following centrality measures for them: Betweenness, Closeness, Authority score and Hub score. Then compare the following groups of metrics, using one plot for each group:

- Betweenness, Closeness and PageRank.
- PageRank, Authority score and Hub score.

Each plot should illustrate the corresponding measures for the 30 nodes ranked by decreasing order of their PageRank.

The reference manuals regarding node centrality, community detection algorithms and graph generators are the following:

https://snap.stanford.edu/snappy/doc/reference/centr.html https://snap.stanford.edu/snappy/doc/reference/con
https://snap.stanford.edu/snappy/doc/reference/generators.html

2.2.1 Answer

Part 1: Identifying execution parameters.

Execution remarks In order to run the test cases included in file *hw1-2.py*, for this part of the assignment, one should execute it in the following fashion:

```
python hw1-2.py TestAlgorithmsMethods.test_find_params
```

Code structure (package SnaHomework1.Part2) The code is structured in the following classes:

- **Algorithms**: implements SNAP function wrappers for the algorithms requested by the assignment, as well as a custom function to compute max node degree (*compute_max_degree*).
- **Util**: contains helper functions for converting SNAP data structures (*TIntFltH*) to arrays, for pretty-printing results and plotting.

Note that code is fully documented inline.

Output

```
-------------------- ITERATION 1 --------------------
```

```
Execution parameters
---------------------------------------------------
                 Parameter                   Value
0          Number of nodes 50.00000000000000000000
1          Node out degree 12.00000000000000000000
2  Node rewire probability  0.57038750344838518025
---------------------------------------------------
```

```
Node with highest degree
---------------------------------------------------
   Node ID  Degree
0       10      29
---------------------------------------------------
```

```
IDs of nodes with the highest Hub and Authority scores, along with their scores
---------------------------------------------------
   Node ID       Type                  Score
0       10        Hub 0.19277612564523194383
1       10  Authority 0.19277612564523191607
---------------------------------------------------
```

```
Execution times of the Girvan-Newman community detection algorithm and the Clauset-Newman-Moore
---------------------------------------------------
              Algorithm            Time
0         Girvan-Newman 00:00:00.437295
1  Clauset-Newman-Moore 00:00:00.000034
2                 TOTAL 00:00:00.437329
---------------------------------------------------
```

```
-------------------- ITERATION 2 --------------------
```

```
Execution parameters
---------------------------------------------------
                 Parameter                    Value
0          Number of nodes 100.00000000000000000000
1          Node out degree   5.00000000000000000000
```

```
2  Node rewire probability   0.57108773398426027068
---------------------------------------------------




Node with highest degree
---------------------------------------------------
   Node ID  Degree
0       50      14
---------------------------------------------------




IDs of nodes with the highest Hub and Authority scores, along with their scores
---------------------------------------------------
   Node ID       Type                  Score
0       50        Hub 0.14351741495686598515
1       50  Authority 0.14351741495698097650
---------------------------------------------------




Execution times of the Girvan-Newman community detection algorithm and the Clauset-Newman-Moore
---------------------------------------------------
              Algorithm            Time
0         Girvan-Newman 00:00:00.869452
1  Clauset-Newman-Moore 00:00:00.000036
2                  TOTAL 00:00:00.869488
---------------------------------------------------


-------------------- ITERATION 3 --------------------


Execution parameters
---------------------------------------------------
                 Parameter                    Value
0          Number of nodes 150.00000000000000000000
1          Node out degree   8.00000000000000000000
2  Node rewire probability   0.91580577075074565130
---------------------------------------------------
```

```
Node with highest degree
---------------------------------------------------
   Node ID  Degree
0       20      25
---------------------------------------------------



IDs of nodes with the highest Hub and Authority scores, along with their scores
---------------------------------------------------
   Node ID       Type                  Score
0       20        Hub 0.12892303359188042600
1       20  Authority 0.12892303359188042600
---------------------------------------------------



Execution times of the Girvan-Newman community detection algorithm and the Clauset-Newman-Moore
---------------------------------------------------
              Algorithm            Time
0         Girvan-Newman 00:00:07.644020
1  Clauset-Newman-Moore 00:00:00.000051
2                 TOTAL 00:00:07.644071
---------------------------------------------------


-------------------- ITERATION 4 --------------------


Execution parameters
---------------------------------------------------
                 Parameter                    Value
0          Number of nodes 200.00000000000000000000
1          Node out degree   5.00000000000000000000
2  Node rewire probability   0.22783821234108858622
---------------------------------------------------



Node with highest degree
---------------------------------------------------
   Node ID  Degree
0      150      15
---------------------------------------------------
```

```
IDs of nodes with the highest Hub and Authority scores, along with their scores
---------------------------------------------------
   Node ID       Type                  Score
0      150        Hub 0.13035279101213675945
1      150  Authority 0.13034971923542587602
---------------------------------------------------




Execution times of the Girvan-Newman community detection algorithm and the Clauset-Newman-Moore
---------------------------------------------------
              Algorithm            Time
0         Girvan-Newman 00:00:03.309928
1  Clauset-Newman-Moore 00:00:00.000057
2                 TOTAL 00:00:03.309985
---------------------------------------------------


-------------------- ITERATION 5 --------------------


Execution parameters
---------------------------------------------------
                 Parameter                    Value
0          Number of nodes 250.00000000000000000000
1          Node out degree   9.00000000000000000000
2  Node rewire probability   0.14947560359323863732
---------------------------------------------------




Node with highest degree
---------------------------------------------------
   Node ID  Degree
0      129      22
---------------------------------------------------




IDs of nodes with the highest Hub and Authority scores, along with their scores
---------------------------------------------------
   Node ID       Type                  Score
```

```
0      215        Hub 0.08087459298888670378
1      215  Authority 0.08087425094244195256
---------------------------------------------------



Execution times of the Girvan-Newman community detection algorithm and the Clauset-Newman-Moore
---------------------------------------------------
              Algorithm            Time
0         Girvan-Newman 00:00:47.858878
1  Clauset-Newman-Moore 00:00:00.000067
2                 TOTAL 00:00:47.858945
---------------------------------------------------


-------------------- ITERATION 6 --------------------


Execution parameters
---------------------------------------------------
                 Parameter                    Value
0          Number of nodes 300.00000000000000000000
1          Node out degree  16.00000000000000000000
2  Node rewire probability   0.93475916956268922942
---------------------------------------------------




Node with highest degree
---------------------------------------------------
   Node ID  Degree
0       96      47
---------------------------------------------------




IDs of nodes with the highest Hub and Authority scores, along with their scores
---------------------------------------------------
   Node ID       Type                  Score
0       96        Hub 0.08566396521351107851
1       96  Authority 0.08566396521351110627
---------------------------------------------------
```

```
Execution times of the Girvan-Newman community detection algorithm and the Clauset-Newman-Moore
---------------------------------------------------
              Algorithm            Time
0         Girvan-Newman 00:02:57.020556
1  Clauset-Newman-Moore 00:00:00.000084
2                 TOTAL 00:02:57.020640
---------------------------------------------------


-------------------- ITERATION 7 --------------------


Execution parameters
---------------------------------------------------
                 Parameter                       Value
0          Number of nodes 350.00000000000000000000
1          Node out degree  12.00000000000000000000
2  Node rewire probability   0.77462770728006313803
---------------------------------------------------




Node with highest degree
---------------------------------------------------
   Node ID  Degree
0      111      33
---------------------------------------------------




IDs of nodes with the highest Hub and Authority scores, along with their scores
---------------------------------------------------
   Node ID       Type                  Score
0      315        Hub 0.07878053907995639926
1      315  Authority 0.07878053907995635763
---------------------------------------------------




Execution times of the Girvan-Newman community detection algorithm and the Clauset-Newman-Moore
---------------------------------------------------
              Algorithm            Time
0         Girvan-Newman 00:03:28.779256
1  Clauset-Newman-Moore 00:00:00.000093
```

```
2                 TOTAL 00:03:28.779349
---------------------------------------------------


-------------------- ITERATION 8 --------------------


Execution parameters
---------------------------------------------------
                 Parameter                    Value
0          Number of nodes 400.00000000000000000000
1          Node out degree  13.00000000000000000000
2  Node rewire probability   0.58000393130526350927
---------------------------------------------------




Node with highest degree
---------------------------------------------------
   Node ID  Degree
0      367      36
---------------------------------------------------




IDs of nodes with the highest Hub and Authority scores, along with their scores
---------------------------------------------------
   Node ID       Type                  Score
0      367        Hub 0.06929466303420397932
1      367  Authority 0.06929466303420361850
---------------------------------------------------




Execution times of the Girvan-Newman community detection algorithm and the Clauset-Newman-Moore
---------------------------------------------------
              Algorithm            Time
0         Girvan-Newman 00:05:18.025574
1  Clauset-Newman-Moore 00:00:00.000103
2                 TOTAL 00:05:18.025677
---------------------------------------------------


-------------------- ITERATION 9 --------------------
```

```
Execution parameters
---------------------------------------------------
                 Parameter                    Value
0          Number of nodes 450.00000000000000000000
1          Node out degree  18.00000000000000000000
2  Node rewire probability   0.98747530073354339297
---------------------------------------------------
```

```
Node with highest degree
---------------------------------------------------
   Node ID  Degree
0      262      54
---------------------------------------------------
```

```
IDs of nodes with the highest Hub and Authority scores, along with their scores
---------------------------------------------------
   Node ID       Type                  Score
0      262        Hub 0.07272684835235007639
1      262  Authority 0.07272684835235011802
---------------------------------------------------
```

```
Execution times of the Girvan-Newman community detection algorithm and the Clauset-Newman-Moore
---------------------------------------------------
              Algorithm            Time
0         Girvan-Newman 00:12:13.687824
1  Clauset-Newman-Moore 00:00:00.000100
2                 TOTAL 00:12:13.687924
---------------------------------------------------
```

```
Execution parameters
---------------------------------------------------
                 Parameter                    Value
0          Number of nodes 450.00000000000000000000
1          Node out degree  18.00000000000000000000
2  Node rewire probability   0.98747530073354339297
```

```
---------------------------------------------------
```

```
.
----------------------------------------------------------------------
Ran 1 test in 1497.778s

OK
```

Part 2: Calculating measures.

Execution remarks In order to run the test cases included in file *hw1-2.py*, for this part of the assignment, one should execute it in the following fashion:

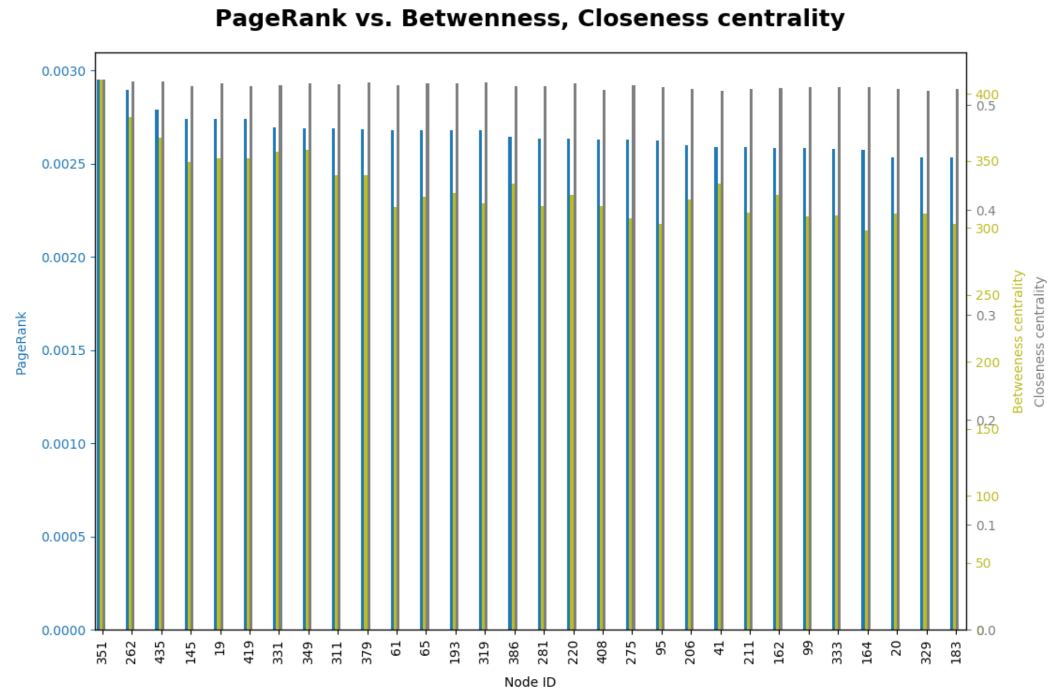
```
python hw1-2.py TestAlgorithmsMethods.test_measures
```

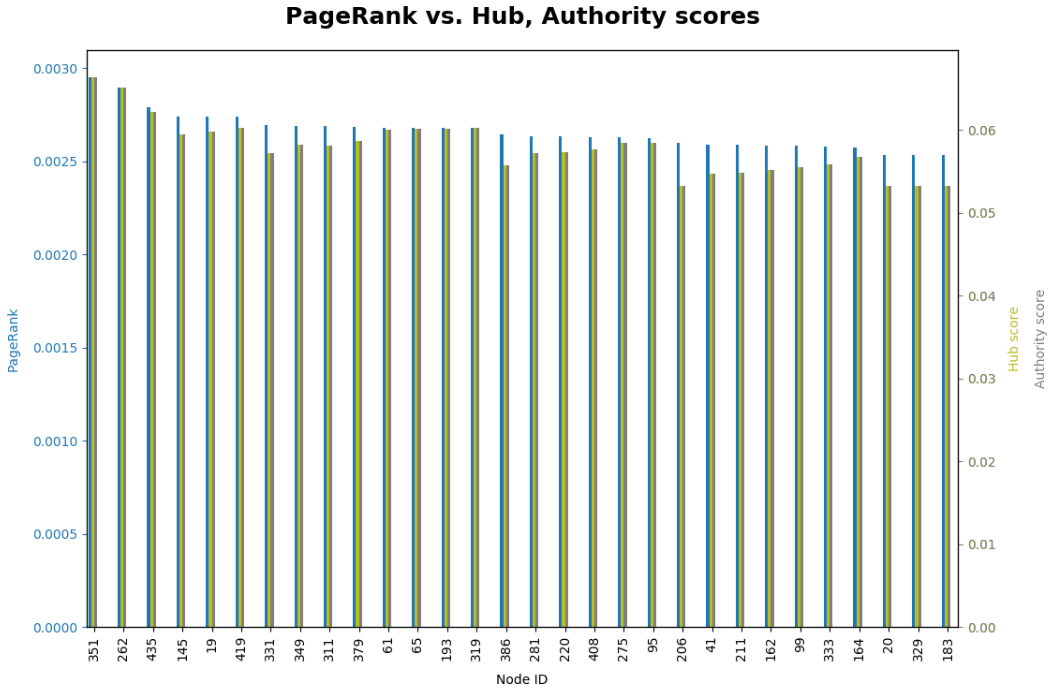
Output The entire Pandas dataframe containing the calculated graph measures is shown below:

```
     page_rank  betweenness_centrality  closeness_centrality  hub_scores  authority_scores
351   0.002950              410.297696              0.523921    0.066332          0.066332
262   0.002895              382.316129              0.522701    0.065179          0.065179
435   0.002790              366.949651              0.522701    0.062221          0.062221
145   0.002743              348.697947              0.517878    0.059512          0.059512
19    0.002741              351.771223              0.520882    0.059851          0.059851
419   0.002740              351.655834              0.517878    0.060216          0.060216
331   0.002695              356.347719              0.519075    0.057193          0.057193
349   0.002691              358.309559              0.520278    0.058252          0.058252
311   0.002690              339.215306              0.519676    0.058070          0.058070
379   0.002686              339.045707              0.521487    0.058700          0.058700
61    0.002679              315.360265              0.519075    0.060034          0.060034
65    0.002679              323.368911              0.520882    0.060201          0.060201
193   0.002679              326.238346              0.520882    0.060189          0.060189
319   0.002678              318.311019              0.521487    0.060278          0.060278
386   0.002643              332.914666              0.517878    0.055690          0.055690
281   0.002634              316.442404              0.517878    0.057188          0.057188
220   0.002633              324.332431              0.520278    0.057319          0.057319
408   0.002631              316.074358              0.514318    0.057638          0.057638
275   0.002628              306.783723              0.518476    0.058501          0.058501
95    0.002627              302.937751              0.516686    0.058477          0.058477
206   0.002597              321.076435              0.515499    0.053241          0.053241
41    0.002587              332.565692              0.513730    0.054721          0.054721
211   0.002587              311.338527              0.514908    0.054874          0.054874
162   0.002587              324.488714              0.516092    0.055133          0.055133
99    0.002584              308.403783              0.517281    0.055564          0.055564
333   0.002581              309.404939              0.517281    0.055857          0.055857
164   0.002576              298.078829              0.517281    0.056785          0.056785
20    0.002537              310.606026              0.515499    0.053288          0.053288
```

```
329   0.002536               310.313070              0.513730    0.053207               0.053207
183   0.002535               302.576136              0.515499    0.053252               0.053252
.
----------------------------------------------------------------------
Ran 1 test in 4.303s

OK
```





Plots

2.3 General remarks

As per the assignment instructions, **Stanford's SNAP** graph library was used for every aspect of the graph creation traversal metrics evaluation procedures.

Furthermore, the following Python packages were used:

- **Numpy**: this package was used as a convenience, in particular for storing the node degree of a graph in function SnaHomework1.Part1.Util.get_in_out_degree_table().
- **Pandas**: this package was used again as a convenience, particularly for representing all graph measures (PageRank, betweenness etc.) in a tabular format and performing sorting and slicing on the data.
- **Matplotlib**: this package was used for producing the plots required.